UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/711,706	09/30/2004	Arash Massoudi	16906.5.1	5705

22913 7590 12/12/2008
Workman Nydegger
1000 Eagle Gate Tower
60 East South Temple
Salt Lake City, UT 84111

EXAMINER	
WU, JUNCHUN	

ART UNIT	PAPER NUMBER
2191	

MAIL DATE	DELIVERY MODE
12/12/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/711,706	**Applicant(s)** MASSOUDI, ARASH	
	Examiner JUNCHUN WU	**Art Unit** 2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 27 August 2008.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-16 and 20-33* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-16 and 20-33* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date *11/21/2008*.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

1. This office action is in response to amendment filed on Aug. 27, 2008

2. Claims 1-16 and 20-33 are pending in this application.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-5, 7, 10, 12-16, 20-25, 27, 29-32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Beisiegel et al (U.S. Pub No. 20040177335 A1 hereinafter "Beisiegel", in view of Skrzynski et al (U.S. Patent No. 6,691,302 B1 hereinafter "Skrzynski") and further view of Fletcher et al. (U.S. Patent No. 7,035,944 B2 hereinafter "Fletcher").

Per claim 1 (Currently Amended)

Beisiegel discloses

A system for construction of a customizable software system and framework, the system comprising:

- a server infrastructure ([0014] *"the deployed service for operation on an application server"*) and a set of management and design tools stored on physical computer-readable media that, when executed by the server infrastructure, causes the system to perform management and development of service-oriented software modules as services ([0015] "*accordance with this services toolkit, the toolkit further comprises a service deployment* *component for deploying the one or more services in accordance with a respective access definition sufficient to access a respective service and the services toolkit is adapted to communicate with an integrated development environment to assist with architecting the enterprise application."*).

But Beisiegel do not disclose

- the system uses software service modules to perform system functions to enable operation of the system itself, wherein execution of the system functions includes the software service modules of the system functions being implemented through the server infrastructure itself

However, Skrzynski discloses

- the system uses software service modules to perform system functions to enable operation of the system itself, wherein execution of the system functions includes the service-oriented software service modules of the system functions being implemented through the server infrastructure itself (col.2 lines 53-62 "*Furthermore, new services may be readily added to the system without changing the dependability system. For example, with the invention a system designer merely has to replace the appropriate DLL or JAVA class with a new service module that implements the desired functionality. The inventive use of the system configuration database (registry) enables system designers to readily <u>implement service functionality with existing service modules</u> regardless of the programming language used to create the service modules.*").
- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify teaching of Beisiegel with the teachings of Skrzynski

to include the system uses software service modules to perform system functions to enable operation of the system itself, wherein execution of the system functions includes the software service modules of the system functions being implemented through the server infrastructure itself in order to provide a method for adding a new service more readily to the system without modifying the service modules.

Both Beisiegel and Skrzynski do not disclose

- service interface definitions for the service-oriented software service modules that perform system functions are first described using the set of management and design tools and then consumed by the same set of management and design tools

But Fletcher discloses

- service interface definitions for the service-oriented software service modules that perform system functions are first described using the set of management and design tools and then consumed by the same set of management and design tools (col.4 lines 12-16 "*In preferred embodiments, this technique comprises: defining a system interface for a collection of one or more software resources; populating the system interface with one or more management functions; specifying the populated system interface in a service description document*" & col.7 lines 35-39 "*According to the present invention, the software resource performing a particular function may be statically bound to a web service proxy, or a web service proxy may be bound to a software resource which is dynamically selected*" In here, software resource refers as web services using the services of a content framework (Abstract) and col.6 lines 26-45 "*One aspect of the present invention also provides a tool for composition or aggregation of new web services.*

Using this composition tool, a systems administrator (or, equivalently, a service composer or other person) may define a new service composed of more fine-grained services...").

- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine teachings of Beisiegel and Skrzynski with the teachings of Fletcher to include service interface definitions for the service-oriented software service modules that perform system functions are first described using the set of management and design tools and then consumed by the same set of management and design tools in order to provide a technique for dynamically integrating software resources in a distributed network (col.3 lines 45-47).

Per claim 2 (Currently Amended)

the rejection of claim 1 is incorporated

Beisiegel discloses

- a communication module between the set of software management and design tools developed for supporting the system functions of the system and a runtime platform of the system through a set of service-oriented software service modules implemented through the same set of management and design tools provided to end-users of the system ([0085] "*Various tools aid a user to create and edit the different aspects of service definitions 902 and to use the definitions 902 to create additional development artifacts 904 and 906 for an application server...*")

- wherein the communication module accomplishes transparent distribution for parts of the system that are consumers of service-oriented software services from parts of the system that are producers of service-oriented software services ([0039] "*Each component is in communication with the other components via a suitable communications bus 206 as required.*").

Per claim 3 (Previously Presented)

the rejection of claim 1 is incorporated

Beisiegel discloses

- the server infrastructure and set of management and design tools configured to perform rapid convergence of quality during construction and advancement of the software system functions through the same set of management and design tools provided to end-users of the system ([0080] *"... used to specify additional quality service attributes for the deployed service"* [0104] "*Advanced transactional connectivity capabilities help developers avoid custom coding by providing extended transactional support for the many challenges related to integrating existing software assets with a J2EE environment.*").

Per claim 4 (Previously Presented)

the rejection of claim 1 is incorporated

Beisiegel discloses

- the server infrastructure and set of management and design tools configured to reduce an implementation time of the system functions through the same set of management and design tools provided to end-users of the system ([0104] "An IDE as adapted by the services toolkit described above and application server help build new applications that integrate with existing assets by leveraging a service-oriented architecture to reduce the complexity of large-scale application development and promote reuse by offering a standard way of representing and interacting with virtually all software assets.").

Per claim 5 (Currently Amended)

the rejection of claim 1 is incorporated

Beisiegel discloses

- the server infrastructure and set of management and design tools configured to create a high-degree of customizability through exposure of system functions as consumable service-oriented software service modules through the same set of management and design tools provided to end-users of the system ([0086] "*Services toolkit 308 provides a graphic user interface (GUI) for working with service definitions having customizable perspectives for presenting views to promote role-based development.*")

Per claim 7 (Currently Amended)

the rejection of claim 2 is incorporated

Beisiegel discloses

- a service manager tool for managing current running services that uses a set of service-oriented software service modules to interact with the system through the same set of management and design tools provided to end-users of the system ([0008] "*a consistent way of interacting with aspects of the enterprise application such as connectors, web services, messaging applications and EJB components*").

Per claim 10 (Currently Amended)

the rejection of claim 2 is incorporated

Beisiegel discloses

- a consumer account provisioning manager tool used to provision and deploy service-oriented solutions that uses a set of software service modules to interact with the system through the same set of management and design tools provided to end-users of the system ([0020] "*deploy the one or more services, each deployed in accordance with an access definition sufficient to access the service.*").

Per claim 12 (Currently Amended)

the rejection of claim 2 is incorporated

Lai discloses

a system cluster manager used for load-balancing and managing clusters of the system that uses a set of service-oriented software service modules to interact with the cluster of systems ([0960] *"Queues may be clustered to provide message-level load balancing and resilience. In other words, the workload may be shared between queues under the same cluster. When the transaction volume goes up, architects and developers may add more queues in the same machine or in another machine."*).

Per claim 13 (Currently Amended)

the rejection of claim 2 is incorporated

Beisiegel discloses

- any management or design tool that needs to interact with the system to use a set of service-oriented software service modules to interact with the system through the same set of management and design tools provided to end-users of the system ([0008] "*a consistent way of interacting with aspects of the enterprise application such as connectors, web services, messaging applications and EJB components*").

Per claim 14 (Currently Amended)

the rejection of claim 1 is incorporated

Beisiegel discloses

- extended through a set of service-oriented software service modules to implement the system functions required for supporting the functionality of the system through the same set of management and design tools provided to end-users of the system ([0084] "*The tool components are grouped and presented conveniently in a service perspective or view, which extends base tool components available in a typical IDE.*").

Per claim 15 (Currently Amended)

the rejection of claim 1 is incorporated

Beisiegel discloses

- all system functions required for management, design and invocation of system functionality by the system are implemented as service-oriented software service modules through the same set of management and design tools provided to end-users of the system ([0082] "*Server component 812 receives and responds to service invocation requests via SOAP over HTTP protocols. Using artifact 808, the SOAP server component determines the particulars for service invocation.*").

Per claim 16 (Currently Amended)

the rejection of claim 1 is incorporated

Beisiegel discloses

- the implementation of all of the system functions that are service-oriented software service modules can be replaced transparently for customizing the system functionality using the same set of management and design tools provided to end-users of the system (col.2 lines 55-57 "*with the invention a system designer merely has to replace the appropriate DLL or JAVA class with a new service module that implements the desired functionality.*").

Per claim 20 (Currently Amended)

Beisiegel discloses

In a computer network, a service-oriented development system for the composition and implementation of service-oriented software modules, the service-oriented development system itself being built on top of service-oriented software modules, the service-oriented development system comprising:

- a user interface tool stored on physical computer-readable media that when executed by one or more processors, causes the system to allow an end-user to develop, assemble, manage and/or execute implementation of service-oriented software modules ([0085] "*Various tools aid a user to create and edit the different aspects of service definitions 902 and to use the definitions 902 to create additional development artifacts 904 and 906 for an application server…*").
- a run-time server having a core module that provides a framework utilizing interfaces with pluggable implementations for dispatching the service-oriented software modules, the core module comprising a hard-coded portion for getting or fetching a definition of any service module ([0016] "*The services toolkit may including one or more tools to assist with the creation of the service definition for each service provider type*" & [0055] "*<u>At the core of the programming model 400 of the services toolkit </u>308 are enterprise services, or services 312 for short. Services 312 are used to model different kinds of service providers 310 in a consistent way. The following is an overview of the programming model 400*").
- each of the user interface tool and run-time server requiring system functions to enable the operation of the service-oriented development system ([0006] "*The J2EE standard wraps and embraces existing resources required by multi tier applications with a unified, component-based application model and enables co-operating components, tools, systems, and applications for solving the strategic requirements of an enterprise.*"),

wherein the core module is configured to implement the at least some system functions by invoking the service modules of the at least some system function ([0083] "*involves creating a Java Bean which may be invoked by a client application (not shown) to access a deployed service…*").

But Beisiegel do not disclose

- some of the system functions of the user interface tool and run-time server are implemented as service-oriented software service modules

However, Skrzynski discloses

- some of the system functions of the user interface tool and run-time server are themselves implemented using service modules (col.2 lines 53-62 "*Furthermore, new services may be readily added to the system without changing the dependability system. For example, with the invention a system designer merely has to replace the appropriate DLL or JAVA class with a new service module that implements the desired functionality. The inventive use of the system configuration database (registry) enables system designers to readily* <u>*implement service functionality with existing service modules*</u> *regardless of the programming language used to create the service modules.*").
- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify teaching of Beisiegel with the teachings of Skrzynski to include the system uses software service modules to perform system functions to enable operation of the system itself, wherein execution of the system functions includes the software service modules of the system functions being implemented through the

server infrastructure itself in order to provide a method for adding a new service more readily to the system without modifying the service modules.

Both Beisiegel and Skrzynski do not disclose

- Using the user interface tool to define definitions of the system functions that are service-oriented software modules.

But Fletcher discloses

- Using the user interface tool to define definitions of the system functions that are service-oriented software modules (col.4 lines 12-16 "*In preferred embodiments, this technique comprises: defining a system interface for a collection of one or more software resources; populating the system interface with one or more management functions; specifying the populated system interface in a service description document*". In here, software resource refers as web services using the services of a content framework (Abstract).
- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine teachings of Beisiegel and Skrzynski with the teachings of Fletcher to include Using the user interface tool to define definitions of the system functions that are service-oriented software modules in order to provide a technique for dynamically integrating software resources in a distributed network (col.3 lines 45-47).

Per claim 21 (Currently Amended)

the rejection of claim 20 is incorporated

Beisiegel discloses

- the user-interface tool comprises a management tool for allowing an end-user to manage service-oriented software modules ([0023] "*Using the service-oriented approach to integration in accordance with the present invention reduces the complexity, cost, and risk of integration by providing a single, simple architectural framework based on web services in which to build, deploy, and manage application functionality.*").

Per claim 22 (Currently Amended)

the rejection of claim 20 is incorporated

Skrzynski discloses

- the at least some of the system functions of the user interface tool and the run-time server being themselves built to use service-oriented software service modules comprises at least one of: service interface metadata management; log analyzing; searching; service monitoring and management; cache management; system configuration; shared memory management; event broadcasting and notification; security management; provisioning; or cluster management (col.4 lines 35-38 "*In one embodiment, each service 66-70 follows a standard procedure and implements a standard interface that allows the <u>service to be monitored</u> and controlled remotely.*").

Per claim 23 (Previously Presented)

the rejection of claim 20 is incorporated

Beisiegel discloses

- the user interface tool and run-time server allow the end-user to customize, replace, or extend one or more of the at least some of the system functions using the end-user functionality of the service-oriented development system ([0086] "… *A service perspective customizes the layout of the GUI to facilitate the development of enterprise services containing the views of various resources for use to develop such services.*").

Per claim 24 (Previously Presented)

the rejection of claim 20 is incorporated

Beisiegel discloses

- The service-oriented development system as recited in claim 20, wherein the user interface tool and the run-time server allow the end-user to rapidly converge a quality of a software system under construction ([0080] *"... used to specify additional quality service attributes for the deployed service"* [0104] "*Advanced transactional connectivity capabilities help developers avoid custom coding by providing extended transactional support for the many challenges related to integrating existing software assets with a J2EE environment.*").

Per claim 25 (Currently Amended)

the rejection of claim 20 is incorporated

Beisiegel discloses

- the user-interface tool consumes those system functions built as service-oriented service modules, and wherein implementation of those services are dispatched by the core module and implemented through the same framework that is provided to the end-user for developing, assembling, managing and/or executing implementation of service-oriented software modules ([0016] "*The services toolkit may including one or more tools to assist with the creation of the service definition for each service provider type*" & [0055] "*<u>At the core of the programming model 400 of the services toolkit</u> 308 are enterprise services, or services 312 for short. Services 312 are used to model different kinds of service providers 310 in a consistent way. The following is an overview of the programming model 400*").

Per claim 27 (Currently Amended)

Beisiegel discloses

In a network environment comprising a service-oriented development system for the composition, management, and implementation of service-oriented software modules, the service-oriented development system including a user-interface tool for allowing an end-user to develop, assemble, manage, and/or execute implementation of service-oriented software modules ([0085] "*Various tools aid a user to create and edit the different aspects of service definitions 902 and to use the definitions 902 to create additional development artifacts 904 and 906 for an application server...*") and including a runtime server for implementing service modules, a method for transparently distributing service invocations of service modules comprising:

- using an invoker interface to request the one or more service-oriented software service modules for performing a system function to enable operation of the service-oriented development system ([0083] "*A service proxy is generated from a service interface for the service to be invoked and an outbound binding that describes how to access the service interface.*"), wherein,

- consuming a particular service module comprises generating an invocation request, and sending the invocation request to a core module ([0082] "*Server component 812 receives and responds to service invocation requests (e.g. remote procedure calls (RPC)) via SOAP over HTTP protocols.*").
- at the core module, receiving the invocation request from the consumer of the service-oriented software service module, the consumer being the service-oriented development ([0082] "*ORB run-time component 814 may receive a RMI-IIOP invocation for session EJB 816 and pass the invocation through.*").
- using a local invoker to access a runtime environment in a same address space as the consumer of the service-oriented software service module ([0083] "*A service proxy is generated from a service interface for the service to be invoked and an outbound binding that describes how to access the service interface.*").
- using a remote invoker to access a runtime environment outside the address space of the consumer of the service-oriented software service module ([0052] "*Access to services is made available via one or more access protocols such as Simple Object Access Protocol (SOAP) run over HTTP and Remote Method Invocation (RMI) run over Internet Inter-Orb Protocol*").
- wherein at least one of the local invoker and remote invoker are used to invoke a service

module for performing a system function of the service-oriented development system ([0052] "*Access to services is made available via one or more access protocols such as Simple Object Access Protocol (SOAP) run over HTTP and Remote Method Invocation (RMI) run over Internet Inter-Orb Protocol*").

wherein operation of the internal invoker and remote invoker are encapsulated from the implementer by the invoker interface such that the consumer is not aware whether the invocation request is being sent via the local invoker or the remote invoker, and wherein the core module can switch between an offline internal invoker mode and one or more servers having remote invokers ([0052]).

But Beisiegel does not disclose

- using a user-interface tool of a service-oriented development system to define interface definitions for one or more service-oriented software service modules for performing a system function to enable operation of the service-oriented development

However, Fletcher discloses

- using a user-interface tool of a service-oriented development system to define interface definitions for one or more service-oriented software service modules for performing a system function to enable operation of the service-oriented development (col.4 lines 12-16 "*In preferred embodiments, this technique comprises: defining a system interface for a collection of one or more software resources; populating the system interface with one or more management functions; specifying the populated system interface in a service description document* & col.7 lines 35-39 "*According to the present invention, the software resource performing a particular function may be statically bound to a web* *service proxy, or a web service proxy may be bound to a software resource which is dynamically selected"*. In here, software resource refers as web services using the services of a content framework (Abstract).

- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine teachings of Beisiegel and Skrzynski with the teachings of Fletcher to include using a user-interface tool of a service-oriented development system to define interface definitions for one or more service-oriented software service modules for performing a system function to enable operation of the service-oriented development in order to provide a technique for dynamically integrating software resources in a distributed network (col.3 lines 45-47).

Per claim 29 (Previously Presented)

the rejection of claim 28 is incorporated

Beisiegel discloses

- the core module is the consumer of the service-oriented software service module and the service-oriented service module is a system function of the core module required to implement functionality of the core module ([0055] "*At the core of the programming model 400 of the services toolkit 308 are enterprise services, or services 312 for short. Services 312 are used to model different kinds of service providers 310 in a consistent way"*).

Per claim 30 (Previously Presented)

the rejection of claim 29 is incorporated

Skrzynski discloses

- the system function comprises at least one of: service interface metadata management; log analyzing; searching; service monitoring and management; cache management; system configuration; shared memory management; event broadcasting and notification; security management; provisioning; or cluster management (col.4 lines 35-38 "*In one embodiment, each service 66-70 follows a standard procedure and implements a standard interface that allows the service to be monitored and controlled remotely.*").

Per claim 31 (Previously Presented)

the rejection of claim 27 is incorporated

- the core module of one instance of the system can access metadata of another remote instance of the system by using a corresponding remote invoker instead of its own internal invoker when consuming metadata data access service interfaces ([0052] "*Access to services is made available via one or more access protocols such as Simple Object Access Protocol (SOAP) run over HTTP and Remote Method Invocation (RMI) run over Internet Inter-Orb Protocol*").

Per claim 32 (Previously Presented)

the rejection of claim 27 is incorporated

- identifying addresses of other instances of the system; and using the remote invoker to broadcast a message to other instances of the system ([0052] "*Access to services is made* *available via one or more access protocols such as Simple Object Access Protocol (SOAP) run over HTTP and Remote Method Invocation (RMI) run over Internet Inter-Orb Protocol*").

5. Claim 6, 8, and 9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Beisiegel in view of Skrzynski, and further view of Bowman-Amuah (U.S. Pub No. 20010052108 A1 hereinafter "Bowman").

Per claim 6 (Currently Amended)

the rejection of claim 2 is incorporated

Both Beisiegel and Skrzynski do not disclose

- a service log manager tool for managing, viewing and analyzing the services dispatched through the system that uses a set of software services to interact with the system.

However, Bowman discloses

- a service log manager tool for managing, viewing and analyzing the services dispatched through the system that uses a set of software services to interact with the system ([2024] "*Such tools include dynamic analyzers and execution logs.*").
- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made combine teachings of Beisiegel and Skrzynski and further include a service log manager tool for managing, viewing and analyzing the services dispatched through the system that uses a set of software services to interact with the system by the teachings of Bowman in order to provide a management tools for log information about program detected, classify, and generate reports ([1333]).

Per claim 8 (Currently Amended)

the rejection of claim 2 is incorporated

Both Beisiegel and Skrzynski do not disclose

- a service cache manager tool for managing cached services within the system that uses a set of service-oriented software service modules to interact with the system.

However, Bowman discloses

- a service cache manager tool for managing cached services within the system that uses a set of software service modules to interact with the system ([2345] "*The tool must run in the platform selected in order to control usage of disk space, main memory, cache, etc*").

Per claim 9 (Currently Amended)

the rejection of claim 2 is incorporated

Both Beisiegel and Skrzynski do not disclose

- a system shared-memory manager tool for managing content of the system shared memory that uses a set of service-oriented software service modules to interact with the system.

However, Bowman discloses

- a system shared-memory manager tool for managing content of the system shared memory that uses a set of software service modules to interact with the system ([0983] "*At this level of integration, tools share a common repository of development objects, design documents, source code, test plans and data.*").

6. Claim 11, 26, 28 and 33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Beisiegel in view of Skrzynski, and further view of Lai (U.S. Pub No. 20050044197 A1 hereinafter "Lai").

Per claim 11 (Currently Amended)

the rejection of claim 2 is incorporated

Both Beisiegel and Skrzynski do not disclose

- a security manager tool used for user and role management that uses a set of service-oriented software service modules to interact with the system.

Lai discloses

- a security manager tool used for user and role management that uses a set of software service modules to interact with the system ([0422] *"The policy server stores access rights and policies that govern the access level of each service component or system by users and by roles. Theses security components may span two or more tiers."*).
- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made combine teachings of Beisiegel and Skrzynski and further include a security manager tool used for user and role management that uses a set of software service modules to interact with the system by the teachings of Lai in order to enable technology for web services and provide a web security framework and design patterns for designing end-to-end web service security.

Per claim 26 (Previously Presented)

the rejection of claim 20 is incorporated

- the core module is configured to dispatch service-oriented software modules implementation using a multi- threaded process abstraction.

But Lai discloses

- the core module is configured to dispatch service modules implementation using a multi-threaded process abstraction ([0040] "*It provides* _multi-layered API abstractions_ *for simple, high-level, business API, and flexible, low-level, generic API. It is the enabling technology for Web Services and peer-to-peer computing in the J2EE.*").
- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made combine teachings of Beisiegel and Skrzynski and further include the core module is configured to dispatch service modules implementation using a multi- threaded process abstraction by the teachings of Lai in order to enable technology for web services and peer-to-peer computing in J2EE ([0040]).

Per claim 28 (Previously Presented)

The rejection of claim 27 is incorporated

Lai further discloses

wherein using a remote invoker to access a runtime environment outside the address space of the consumer of the service module further comprises:

- serializing the invocation request for the service module; communicating the serialized invocation request using a network protocol; receiving a serialized response including outputs related to the service module for which the invocation request was serialized

([0566] "*The JMS bridge component 1 then serializes the content into a SOAP message and sends it to JMS bridge component 2. There may be a built-in control mechanism to ensure the SOAP messages are resent if JMS bridge component 2 does not receive the content under the same JMS bridge.*").

- deserializing the serialized response to a native object form of the requesting service module ([0567] "*The JMS bridge component 2 deserializes the SOAP message and publishes to another JMS topic…*").
- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine teachings of Beisiegel and Skrzynski and further include foregoing descriptions by the teachings of Lai in order to provide a method for delivery of the message to the subscriber, the subscriber may return an acknowledgement receipt to the bridge. ([0567]).

Per claim 33 (Previously Presented)

The rejection of claim 32 is incorporated

Lai further discloses

- including in the broadcast message a callback invoker address, such that the other instances of the system can return an invocation of a service to the core module ([0733] "*Design patterns for managing distributed events using a Distributed Event-based Architecture (DEBA) have been identified. DEBA patterns make use of state machine concepts and callback design patterns (visitor and observer)*").

- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine teachings of Beisiegel and Skrzynski and further include the broadcast message a callback invoker address, such that the other instances of the system can return an invocation of a service to the core module by the teachings of Lai to include in order to provide an easy implementation of dynamic workflow model ([0733]).

Response to Arguments

Applicant's arguments filed on Aug. 27, 2008 have been fully considered but they are not persuasive.

- In the remarks, Applicant argues that:

In regard to independent claims 1, 20 and 27 applicant respectfully submits currently amended claims that cited references do not disclose or suggest.

Examiner's response:

Examiner disagrees.

Applicant's arguments with respect to claims 1, 20 and 27 have been considered but are moot in view of the new ground(s) of rejection - see Beisiegel, Skrzynski and Fletcher, arts made of record, as applied hereto.

Conclusion

7. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to JUNCHUN WU whose telephone number is (571)270-1250. The examiner can normally be reached on 8:00-17:00 M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

JW

/Wei Y Zhen/
Supervisory Patent Examiner, Art Unit 2191